Table of Contents

Under Investigation

Slow Response to the "ls -l" Command

***Problem**: Users have experienced a slow response to the Unix command ls -l, often waiting for minutes for the command to complete.*

Status: In Progress

Actions:

Updated 08.31.11 - NAS systems staff have deployed a second new filesystem (nobackupp2). In addition, they completed the transition of users to nobackupp1 on August 8th.

Updated 07.27.11 - NAS systems staff have deployed a new filesystem (nobackupp1) with a new RAID controller, which improves the IOPS performance. Users on nobackupp10 are being transitioned to the new filesystem. Users on other nobackup filesystems will be transitioned to similar, new RAID controllers.

Updated 06.28.11 - NAS systems staff are testing new RAID controllers, which will improve Input/Output Operations Per Second (IOPS) performance, and result in an improved stats-per-second rate.

Updated 05.26.11 - NAS systems staff began an immediate investigation, and determined the source of the problem was related to access of the metadata on the Lustre filesystem, and took the following actions:

- Implemented a change to preload metadata information into the buffer cache on all of the nobackup filesystems. This improved the stats-per-second rate.
- Evaluated the solid-state drives (SSDs) to improve performance of stat calls. There are two use cases for accomplishing this: one approach is to use SSDs for filesystem journals, the other is to use them for filesystem metadata. SSDs were deployed on nobackupp10 for the filesystem journal.
- Reduced the default stripe count to 1 and the default stripe size to 4 MB. This change resulted in fewer stat calls and, over time, improve the ls -l response. (Effective January 13, 2011).
- Developed a modified gnu tar command, /usr/local/bin/mtar, which is Lustre stripe aware and will create a tar file or extract files with an appropriate sized stripe count.

Tips:

Updated 08.02.11 - Since the default stripe count was set to 1 to improve overall performance of Lustre, users need to keep in mind that they sometimes should use larger stripe counts, particularly with large files.

- You can set the stripe count on a directory or create a file with a particular stripe count by using the command *lfs setstripe*.
- We recommend using the *mtar* command in place of *tar* whenever you create or extract from a tar file on Lustre, as *mtar* will automatically choose appropriate stripe counts.

See more information on Lustre striping in the articles Lustre Basics and Lustre Best Practices.

Background:

Previously, by default, the Lustre filesystem striped data across object storage targets (OSTs) in 1-MB chunks. So, a 4-MB file would be spread across four OSTs, resulting in four stat calls to get the size of that one file.

MPI Program Fails or Hangs

***Problem**: MPI program fails or hangs due to network communication problems.*

Status: In Progress

Actions:

NAS system staff are monitoring for errors in the InfiniBand fabric, and replacing bad or unreliable cables when detected.

Tips:

If your MPI job aborts or hangs due to InfiniBand problems, your PBS output file will produce error messages similar to the below:

```
MPI ERROR: 14:34:10: rank 960: r199i0n2 IB board mlx4_0 port 1

           had fault with communications to r190i0n6, restarting...
```

In this case, we recommended that you do the following:

1. Wait a few minutes and resubmit your job.
2. File a ticket with the NAS Control Room to report the problem.
3. If you have not done so, use SGI's MPT versions 2.0.4 or later, which are more robust against these InfiniBand problems and provide more diagnostic information in the system log files.

Background:

The network backbone of Pleiades comprises a pair of InfiniBand fabrics that are the currently the largest in the world (for details, see Network Resources). Most of the time, the large number of switches and cables works well, but sometimes, a cable will go bad, or its connection will work loose, causing some data packets to be lost or corrupted. When one cable fails, packets get re-routed, putting additional load on other paths, which can result in congestion and dropped packets.

The Lustre and TCP/IP protocols generally handle these failures by detecting bad or missing packets and retrying. The various MPI implementations cope less well, and with different degrees of success.

So, if your MPI program aborts with an error message that suggests some node had

communication problems with another node (see above) or if the program hangs after issuing such an error message, then the program might have been affected by a cable failure.

Be aware that some communication errors are not caused by bad hardware - one rank running out of memory can cause communication error messages from surviving ranks.

While we monitor for errors in the InfiniBand fabric, and replace bad or unreliable cables when we detect them, paradoxically, the act of replacing a cable can cause its own errors. Lately, there has been an increase in cables needing work, possibly as a result of the recent facility over-heating incident in early April 2011.